

3. (Twice Amended). A method of providing a dynamically configurable operating system on a computer having a working memory, comprising:

providing demand-loadable operating system components initially stored outside of said working memory, each operating system component having an entry point comprising a constructor for an object, and

providing a Namespace in said working memory which provides in said working memory access to ones of said operating system components as they become needed by applications running in said computer, said Namespace managing demand-loading and unloading of said operating system components in said working memory.

5. (Amended). The method of Claim 3 further comprising providing applications in said working memory which rely on said Namespace to furnish access to ones of said operating system components in said working memory as they become needed by ones of said applications.

6. (Amended). The method of Claim 5 wherein each operating system component comprises an object, and wherein the step of providing each demand-loadable component comprises:

providing an IUnknown interface in the object having the following methods:

- (a) add reference for incrementing a count of the number of applications requiring the object;
- (b) release reference for decrementing a count of the number of applications requiring the object; wherein said Namespace is responsive to said count in said managing of said demand-loading and unloading.

7. (Amended). The method of Claim 5 wherein each operating system component comprises an object, and wherein the step of providing a demand-loadable operating system component comprises:

providing an IUnknown interface in the object having a QueryInterface method of providing access to the methods of the object to an application invoking QueryInterface.

9. (Amended). A method of operating a computer having a working memory wherein applications and objects may be loaded during run time, comprising:

providing a Namespace in said computer;  
running an application in said computer;  
said application calling a demand-loadable object by causing the name of said object to be presented to said Namespace, wherein said object is an operating system component;  
in response to being presented with the name of said object, said Namespace returning to said application an IUnknown pointer of said object;  
upon return of said IUnknown pointer of said object, said application using said IUnknown pointer to call a QueryInterface method of said object and request a pointer to a desired interface;  
said QueryInterface method returning said desired interface, whereby said application can invoke a desired method through said interface.

---

13. (Amended). A computer having a working memory and access to a storage memory, said computer comprising:

an application capable of being loaded in to said working memory and running in said computer;

at least one object initially stored in said nonworking memory, said object comprising an operating system component;

a Namespace in said working memory;

said application being programmed to cause said one object to be identified to said Namespace whenever said application finds a need for said object during the running of said application;

said Namespace being programmed to:

(a) respond to said application identifying said one object by determining whether said one object is currently registered in Namespace, and if it is not registered, then,

(b) causing said one object to be loaded from said storage memory to said working memory and,

(c) registering said one object in said Namespace,

(d) upon said object being registered in said Namespace, returning to said application a pointer to said object.

---

18. (Amended). A computer having a working memory and access to a storage

memory, said storage memory holding at least one object, said computer comprising:

an application in said working memory, said application needing access to said one object at a particular time during the running of said application, said object comprising an operating system component, and said application being programmed to cause a request for said one object to issue contemporaneously with said particular time;

a Namespace which is programmed to respond to a request from said application for said one object by loading said one object from said storage memory into said working memory and then providing said application with a pointer to said one object.

---

24. (Twice Amended). A system residing in a working memory of a computer and in a storage memory, said system comprising:

C<sup>6</sup> demand-loadable operating system components initially stored in said storage memory, each operating system component having an entry point comprising a constructor for an object, and

a Namespace in said working memory which provides in said working memory access to ones of said operating system components as they become needed by applications running in said computer, said Namespace managing demand-loading and unloading of said operating system components in said working memory.

---

26. (Amended). The system of Claim 24 further comprising applications in said working memory which rely on said Namespace to furnish access to ones of said operating system components in said working memory as they become needed by ones of said applications.

C<sup>7</sup> 27. (Amended). The system of Claim 26 wherein each component operating system comprises an object, and further comprising:

an IUnknown interface in the object having the following methods:

(a) add reference for incrementing a count of the number of applications requiring the object;

(b) release reference for decrementing a count of the number of applications requiring the object; wherein said Namespace is responsive to said count in said managing of said demand-loading and unloading.

In re Appln. of Forin et al.  
Application No. 09/282,238

28. (Amended). The system of Claim 27 further comprising:  
an IUnknown interface in the object having a QueryInterface method of providing  
access to the methods of the object to an application invoking QueryInterface.

---

**Cancel claim 4, 25.**